Available Models

EM38A: no internal relays.
EM38A-R: 8 relays for simple control.
EM38A-X: 8 relays for complex control.

Technical Specifications

Operation Mode
playback only

Sound File Format
MP3 (ISO 11172-3)

Max. Number of Sound Files
Parallel / Direct Mode: 8
Parallel / Binary Mode: 128
Parallel / Sequential Mode: 8 x 99
Parallel / Round-Robin Mode: 8
Parallel / Script Mode: 999
Serial Port: 999

Memory Card Type
SD/SDHC

Max. Memory Capacity
2 GB for SD (FAT/FAT16)
32 GB for SDHC (FAT32)

Max. Recording Time (w/2GB card)
33 hours (based on 128 kbps bit rate)

Supply Voltage
12 ~ 30 VDC

Typical Standby Current
60 mA

Audio Output
(30V supply, 8 Ohm load, 10% THD+N)
High efficiency class D
Stereo: 15W per channel
Mono: 55W bridge tied load (BTL)

Serial Interface
RS-232 / RS-485

Trigger Inputs
8 inputs, contact closure or 3.3V/5V logic

Physical Dimensions
EM38A
5.3" x 4" x 1.35"
EM38A-R & EM38A-X
5.3" x 6.7" x 1.35"

Inputs, Outputs & Controls

Power Light (PWR)
The power light is turned on when power is applied.

Trigger Inputs Terminals: T1 - T8, GD
The trigger inputs terminals are enabled by default, but disabled automatically if the serial port is enabled. Both dry contact closure and 3.3V/5V logic signal are acceptable. Trigger inputs may work differently depending on the particular trigger mode described later. The GD terminal is ground, connected internally to the power ground.

Reset Input Terminal: RS
Connect this input to the ground momentarily to reset the unit. Min. duration is 100 ms.

Busy Output Terminal: BY
This open collector output from an internal transistor can be controlled with more flexibility if the unit runs in the Script mode. In all other modes it is automatically turned on while playing audio, with a maximum sink current of 200 mA. It can be used to synchronously activate an external relay for switching on a device such as a lamp or a motor.

Power Input Terminals: V+, GD
Use a well regulated DC power supply to obtain the best sound quality. Connect the power supply's positive output to terminal V+, and the negative output to terminal GD. Alternatively, power can also be supplied via the 2.1mm center positive coaxial jack located on the left side of the unit.

Line Output (LINE): 1/8" Stereo Phone Jack
This jack provides single ended line output.

Balance Pot (BAL)
This pot adjusts the output balance between the two channels. It should be set at the middle (center detent) if the unit is configured for BTL (bridge tied load) mono out.

Volume Pot (VOL)
Turn this pot clockwise to increase the output volume. It affects both the speaker and the line out.

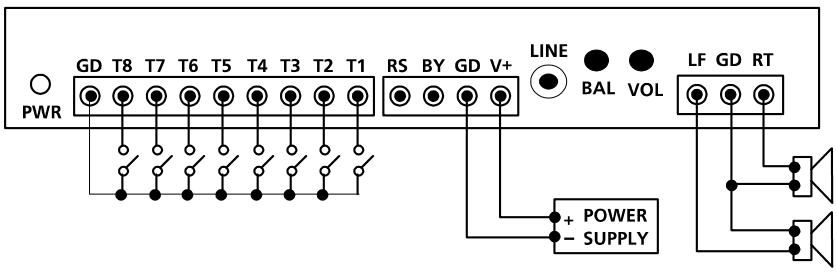
Speaker Output Terminals: LF (left channel), GD, RT (right channel)
See the Speaker Connections section.

Serial Port (DB9 Female)
The default setting is RS-232. To select RS-485, move the internal jumper JP1 to the "485" setting. The serial port is disabled by default. To enable the serial port, you must create a configuration file (MODE.TXT) on the flash card as described later. The trigger inputs are disabled automatically when the serial port is enable.

DC Adaptor Jack
Instead of the power input terminals, power can also be supplied via this 2.1mm center positive coaxial jack located on the left side of the unit.

Relay Output Terminals (EM38A-R & EM38A-X only)
Three terminals are provided for each relay: NC (Normally Closed), NO (Normally Open) and Common. The contact is rated at 12A/120VAC or 10A/24VDC.

Typical wiring diagram for push button activation



Parallel Trigger Modes

Parallel Trigger Mode defines how the playback is to be triggered via the trigger inputs. All inputs are internally pulled up and, if left unconnected, have a voltage of 3.3V (logic "1"). 5V signal is tolerated and also seen as logic "1". Logic "0" is ground.

Direct Trigger (default)
In this mode each input directly triggers a corresponding file: T1 = File 001, T2 = File 002, ..., T8 = File 008.

A trigger is valid when the input is shorted to the ground for at least 50 ms. The Direct Trigger is prioritized from T1 (the highest) to T8 (the lowest). However, it does not mean a higher priority input can interrupt a lower one. It only means that if multiple triggers are applied at the same time, the highest priority wins.

Binary Trigger
Use the Binary Trigger to trigger up to 128 different files ranging from 001 to 128. The Binary Trigger is often preferred when the system is controlled by an external controller because it requires less I/O ports in most cases.

To trigger a particular file, the first step is to signal the file number on T1 (LSB) ~ T7 (MSB). The signal must be in the binary format with +5V being logic "1", and 0V (the ground) being logic "0". For example, to signal File 007 (binary code "0000111"), T1 ~ T3 should be at +5V, and T4 ~ T7 should be at 0V. Note that, as a special case, the binary code for File 128 is "0000000".

The second and the last step is to short T8 to the ground for at least 50 ms while maintaining the signal on T1 ~ T7. Afterwards, signals on T1 ~ T7 don't matter any more.

Sequential Trigger
Use the Sequential Trigger to sequentially trigger up to 99 different files per input as described below. However, the total number of files cannot exceed 511 unless the memory card is formatted with the FAT32 file system.

T1 triggers File 001 ~ 099
T2 triggers File 201 ~ 299
.....
T8 triggers File 801 ~ 899

Each trigger on the same input activates the next file in the sequence. The sequence automatically restarts when either the end of the sequence is reached or there is a break in the sequence. For example, if there are only three files on the flash card: 001, 002, and 004, the system will only sequence from 001 to 002. File 004 will never be played because File 003 is missing.

The Sequential Trigger is prioritized from T1 (the highest) to T8 (the lowest). However, it does not mean a higher priority input can interrupt a lower one. It only means that if multiple triggers are applied at the same time, the highest priority wins.

Round-Robin Trigger
This mode is very similar to the Direct Trigger mode except that the inputs are not prioritized. So if multiple inputs are tied to ground then their files will be played one after another, instead of just the highest priority one. Round-Robin mode can only be used in conjunction with Non-interruptible Playback and Script Playback.

Parallel Playback Modes

Non-interruptible Playback (default)
The file is played once per trigger. The playback is not interruptible except by the system reset. Looping is possible by applying a constant trigger on the input.

Interruptible Playback
The file is played once per trigger if not interrupted. Any input but itself can interrupt the playback. Looping is possible by applying a constant trigger on the input.

Holdable Playback
The file is played for as long as the input is triggered, looping if necessary. It is not interruptible except by the system reset.

Script Playback
Please see the Script Playback Mode section for descriptions.

Parallel Trigger Polarity

Close Contact
Input is continuously triggered when it's at 0V (ground).

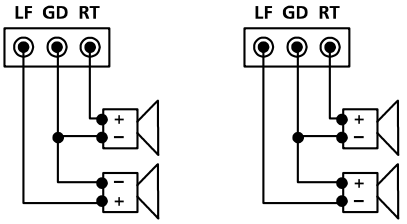
Open Contact
Input is continuously triggered when it's left open or at 3.3V/5V.

Make Contact
Input is triggered one time as it goes from 3.3V/5V to 0V.

Break Contact
Input is triggered one time as it goes from 0V to 3.3V/5V.

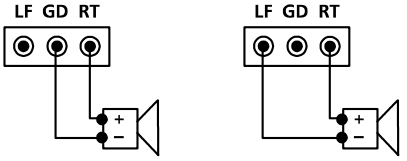
Speaker Connections

Regular Stereo ***Virtual Surround Stereo***



The left channel signal is internally inverted and that's why the left speaker has an inverted polarity for regular stereo output. If the left speaker is not inverted then the output is virtual surround stereo.

Regular Mono ***BTL Mono (4X Output Power)***



Since BTL mono provides four times the output power of regular mono at the same supply voltage, it is often used to boost the output power when the supply voltage is low. However, the speaker impedence should not be lower than 8 Ohms to avoid overloading the power amp.

Script Playback Mode

Instead of playing a single sound file, the Script Playback mode executes a script of commands for each trigger. Note that the Busy output will not turn on/off automatically in the Script mode. It must be specifically turned on/off with the BN and the BF commands.

Written in the configuration file using plain text, the script consists of multiple lines each containing the commands for a particular trigger in the following format:

?nnn=Command1,Command2...

Here "nnn" is the trigger number and "?" is one of the following:

N - Non-interruptible
Execution of this trigger is not interruptible.

I - Interruptible
Execution of this trigger can be interrupted by any trigger other than itself.

H - Holdable
Execution of this trigger continues for as long as the trigger is applied, repeating itself if necessary. The execution stops immediately when the trigger is removed.

In the DS and RS modes, there are 8 direct triggers (001 ~ 008, corresponding to T1 ~ T8). In the BS mode, there are 128 direct triggers (001 ~ 128, corresponding to the binary code on T1 ~ T7).

In addition to the direct triggers, there are also indirect triggers. An indirect trigger can be activated only by jumping from another trigger using the Jump command.

In the DS and RS modes, there are 991 indirect triggers (009 ~ 999). In the BS mode, there are 871 indirect triggers (129 ~ 999).

These are the script commands:

Fnnn - play File #nnn
Example: F168 plays file #168.

Wnnnnn - wait nnnnn units of 0.1 second
Maximum value for nnnnn is 65535 (6553.5 seconds).
Example: W00020 = wait 2 seconds.
Note: W00000 = wait forever.

Jnnn - jump to trigger #nnn
Example: J007 jumps to trigger 007.

BF - turn off the Busy output
Use this command to turn the Busy output off.

BN - turn on the Busy output
Use this command to turn the Busy output on.

XNn - turn on relay #n (EM38A-R & EM38A-X only)
Example: XN8 turns on relay #8.

XNN - turn on all relays (EM38A-R & EM38A-X only)
Example: XNN turns on all relays.

XFn - turn off relay #n (EM38A-R & EM38A-X only)
Example: XF6 turns off relay #6.

XFF - turn off all relays (EM38A-R & EM38A-X only)
Example: XFF turns off all relays.

Xnnn - send binary to relay (EM38A-R & EM38A-X only)
Example: X63 turns on relay #1~6 because 63 = binary 00111111.

END
Always add the word END at the end of the entire script. You may add any comments for your own reference after END.

Important Notes

- All command letters must be in upper case.
- Script lines must be separated by carriage returns (the Enter key).
- A script line is limited to 128 characters, excluding '=' and ','. If more space is needed, use the Jump command.

Automatic Execution of Script 000

Upon powerup or reset, the system will automatically executes script 000 once if it exists.

Script Examples

```
DS
N001=F007,W00030,BN,F899,BF,J168
I168=F001,W36000,J168
H033=F273
END
```

DS is not really a script command, but it tells the system to enter the Direct Script mode. You can also use RS or BS to enter the Round-Robin Script mode or the Binary Script mode.

When the T1 input is triggered, the system start executing trigger N001. Since this trigger is non-interruptible, it will always executes to the end. Trigger N001 is executed as the following:

- play file #007,
- wait 3 seconds,
- turn on the Busy output,
- play file #899,
- turn off the Busy output,
- jump to trigger 168 (I168).

Trigger I168 is executed as the following:

- play file #001,
- wait 60 minutes,
- jump back to itself.

Since trigger I168 is interruptible, this endless loop can be broken by any future trigger.

Trigger H033 will never be executed because it is an indirect trigger in the DS mode, yet it is not jumped to by any other trigger.

Background Music Example

The automatic execution feature can be used to play background music while no trigger is being executed. For example,

```
DS
I000=F123,J000
N001=F001,J000
N002=F002,J000
N003=F003,J000
END
```

Here file #123 is looped from power-up but can be interrupted by trigger inputs T1, T2, and T3. After interruption, it always jumps to trigger 000 so file #123 starts to loop again, although from the beginning instead of where it left off.

System Configuration File

By default, the system works in the following mode without a configu ration file:

Trigger Input: Parallel
Trigger Mode: Direct
Playback Mode: Non-Interruptible
Trigger Polarity: Close Contact

To operate the system in any other modes, you need to create a plain text file called "MODE.TXT" with a line of three letters:

First Letter: Trigger Mode
D = Direct
B = Binary
S = Sequential
R = Round Robin

Second Letter: Playback Mode
N = Non-interruptible
I = Interruptible
H = Holdable
S = Script

Third Letter: Trigger Polarity
C (or no letter) = Close Contact
O = Open Contact
M = Make Contact
B = Break Contact

To enable the serial port (RS-232/RS-485), instead of the three letters described above, put a two-digit number ranging from 01 to 99. This number becomes the address of this unit on the serial bus and therefore should be unique.

For Script modes, enter the script starting from the second line. Be sure to add the word END at the end of the Script.

After editing the configuration file, be sure to save it as a "plain text file", "ASCII text file", or simply "text file". The system may not work if the configuration file is not created properly.

File Number Assignment

Sound files on the flash card must be assigned a unique file number for identification purpose. The file number must be a three digit number within the following range:

For Direct Trigger: 001 ~ 008
For Binary Trigger: 001 ~ 128
For Sequential Trigger and Serial Control: 001 ~ 999

Simply add the file number to the beginning of the original filename, e.g. "001 tiger.mp3". Note that if you want to store the maximum number of files on the flash card, you should keep the filenames (including the file number) within 8 characters and use numbers and capital letters only.

Relay Control (EM38A-R / EM38A-X only)

The relays are controlled differently in different modes:

QSA mode (EM38A-X only)
Relays are pre-programmed for synchronized activations during audio playback by using the Windows based, user friendly QSAplay program. A PC is needed for programming only, but not for the actual operation.

Serial port mode
Relays are controlled by commands received via the serial port. Please see the Serial Commands section for details.

Script mode
Relays are controlled by script commands. Please see the Script Mode section for details.

All other modes
A relay will turn on when the corresponding file is being played. Relay #1 corresponds to file #001, relay #2 corresponds to file #002, and so on. For example, relay #8 will be turned on when file #008 is being played.

Trouble Shooting Guide

1. No sound.
a. File numbers are not assigned properly.
b. The system is in the wrong mode due to missing or incorrect configuration file.
c. If the flash card is inserted when the power is on, the system may not work. To fix this problem, recycle the power or use the RS input to reset the system.
d. The output volume may have been set too low. Try turning it up.

2. Plays the wrong file.
a. File numbers are not assigned properly.
b. The system is in the wrong mode due to missing or incorrect configuration file.

3. Noisy playback.
The speed of the flash card is too slow. Use a faster flash card or convert the file to a lower bit rate.

Serial Interface

When the serial interface is enabled, all parallel inputs T1 ~ T8 are disabled and all Parallel Modes are no longer available.

The serial interface consists of a female DB9 serial port supporting both RS-232 (default setting) and RS-485. To support RS-485, you must move the internal jumper JP1 to the "485" setting.

On the DB9 connector, three pins are used for RS-232: pin 2 for RX, pin 3 for TX, pin 5 for ground. Two pins on the same connector are used for RS-485: pin 1 for negative, pin 9 for positive.

The hardware protocol is fixed at 9600 baud, eight data bits, no parity and one stop bit (9600, 8N1). Other protocols may be supported by special request.

For RS-232 applications, the system (a DCE device) is connected to a DTE device (such as a PC) with a regular serial cable (not a null modem cable). For RS-485 applications, up to 32 systems, each assigned with a unique address, can be daisy chained on the same bus with a 2-wire cable.

To enable the Serial Mode on the system, the configuration file must contain a two-digit code ranging from 00 to 99. These two digits are the only text required for the configuration file. If the code is 00, the system enters the RS-232 mode. If the code is within 01 to 99, the system enters the RS-485 mode and uses the code as its address on the RS-485 bus.

Both the RS-232 mode and the RS-485 mode use the same software protocol except that in the RS-232 mode, the address byte is not used (see below).

The communication protocol uses software handshake on a per-byte basis. That is, for every byte it receives, the system sends an confirmation byte to the DTE. The DTE must not send the next byte until it receives the confirmation, otherwise commands may not be received properly.

A communication session always starts with an initial process. The DTE always initiates the communication by sending an ASCII 'A' followed immediately by the binary code for the RS-485 address. For the RS-232 mode the address code is not used/sent. The device should respond by sending back an ASCII "a" within 100 ms. Otherwise the initial process has failed and should be restarted.

After the initial process the DTE can issue the command one byte at a time. For each byte sent, the DTE should wait to receive a proper confirmation within 100 ms. If the confirmation is missing or invalid, the whole session must be aborted.

The initial process must precede each and every command.

Testing the Serial Port

The serial port can be easily tested by using a Windows utility program called "HyperTerminal". HyperTerminal allows you to send and receive data through the PC's serial port. All you need to do is type the letters on the keyboard to send them, and watch the screen for received data.

The first step is to create a text file called "MODE.TXT" on the flash card. Put only two letters in the file: 66. This will put the player into the RS-485 mode with an address of 66. You may use Windows Notepad or Wordpad to create the file, just be sure to save the file as "text only".

In addition to the MODE.TXT file, you also need to put a test sound file on the flash card. For testing purpose, this sound file should be called "001.mp3".

The next step is to connect the player to the PC using a "straight through" serial cable, not a "null modem" one. A straight through cable has pin 2 connected to pin 2, pin 3 connected to pin 3, and pin 5 connected to pin 5. Those are the only three pins we use. Other pins, whether connected or not, do not matter.

Power up the player and run HyperTerminal on the PC. On Windows XP this program is found under

Start>All Programs>Accessories>Communications

Create a new connection on the COM port to which the player is connected, with the following parameters:

Bits per second = 9600
Data bits = 8
Parity = None
Stop bits = 1
Flow control = None

Now we are ready to send the following commands to the player.

We Type	*Player Responds*	
A	*-*	
B	*a*	*(see notes below)*
F	*f*	
0	*0*	
0	*0*	
1	*1*	

At this point the player should start playing the test sound file, and the test is considered successful.

Note that we typed 'B' as the RS-485 address code because the binary value of ASCII 'B' is 66 - the number we put in the MODE.TXT file earlier. If we had used 00 in the MODE.TXT file earlier to put the system into the RS-232 mode, then this 'B' should not be typed.

If you made a mistake and typed a wrong letter during the test, the player will either respond with a letter 'e' (for error) or not respond at all, depending on the situation. In this case, you need to re-type the command from the very beginning.

Serial Commands

The communication protocol uses software handshake on a per-byte basis. That is, for every byte it receives, the system sends an confirmation byte to the DTE. The DTE must not send the next byte until it receives the confirmation byte.

Play File
DTE Sends: F### (### is the three-digit file number)
System Confirms: f### (### is the same file number as above)

If the file exists, it will be played once. If the file does not exist, the system will simply ignore the command.

If the system is playing/paused when the 'F' letter is received, it will return the error code 'e' instead of 'f'. At this point the command should be aborted. You should use the *Stop Playback* command to stop the currrent playback first before starting a new one.

Loop File
DTE Sends: L### (### is the three-digit file number)
System Confirms: l### (### is the same file number as above)

If the file exists, it will be played repeatedly. If the file does not exist, the system will simply ignore the command.

If the system is playing/paused when the 'L' letter is received, it will return the error code 'e' instead of 'l' (lower case L). At this point the command should be aborted. You should use the *Stop Playback* command to stop the currrent playback first before starting a new one.

Stop Playback
DTE Sends: S
System Confirms: s

If the system is not playing/paused, it will simply ignore the command. Otherwise it will terminate the current playback.

Pause Playback
DTE Sends: P
System Confirms: p

If the system is not playing, it will simply ignore the command.

When the system is being paused, its Busy output (terminal BY) is still active.

Resume Playback
DTE Sends: R
System Confirms: r

If the system is not paused, it will simply ignore the command.

Busy?
DTE Sends: B
System Confirms: b (if playing/paused) or s (otherwise)

Query Parallel Input Status
DTE Sends: Q
System Confirms: one status byte read from the parallel input

This command is usually used when the parallel inputs are connected to switches, sensors and/or other similar devices, and you want to know the on/off status of these devices. Note that, although the status of these devices can be read, they cannot be used to trigger the audio because the Parallel Interface is disabled when the Serial Port is enabled.

The status byte has T1 as the LSB and T8 as the MSB. Note that an input is at logic "1" when connected to +5V or left unconnected, and it is at logic "0" when connected to 0V (the ground).

This command can be issued at any time.

Script Command
DTE Sends: C###C (### is one line of script commands)
System Confirms: c###c (### is an echo of the script line)

For example, if you send "CF001W00020F002C" then the system would confirm with "cf001w00020f002c" as the characters are being received. Upon receipt of the second 'C', the system would execute this script line as described below: play file 001 (F001), wait 2 seconds (W00020), play file 002 (F002).

Refer to the 'Script Playback Mode' section for details on how to write the script.

Relay On (EM38A-R & EM38A-X only)
DTE Sends: XN# (# is the relay number)
System Confirms: xn# (# is the same relay number as above)

For example, "XN3" turns on relay #3. To turn on all relays, use "XNN". This command can be issued at any time.

Relay Off (EM38A-R & EM38A-X only)
DTE Sends: XF# (# is the relay number)
System Confirms: xf# (# is the same relay number as above)

For example, "XF2" turns off relay #2. To turn off all relays, use "XFF". This command can be issued at any time.

Send Binary to Relay (EM38A-R & EM38A-X only)
DTE Sends: Xnnn (nnn is a 3-digit number with leading zeros)
System Confirms: xnnn (nnn is the same number as above)

For example, "X005" turns on relay #1 and #3 and turns off all others because the binary code for 005 is 00000101.

Serial Command Error Code

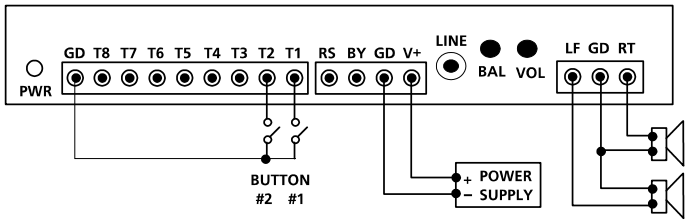
The system will confirm with an "e" if an invalid command is received, or if a valid command is received at the wrong time. At this point, the command should be aborted.

Trouble Shooting Guide

1. The DTE device receives strange characters.
Make sure the DTE device's serial port setting is 9600 baud, 8 data bits, no parity, 1 stop bit.

Application Example

Using Normally Open Push Buttons



1. Play without interruption.

Intended Operation
- Press button #1 to play file #001 once.
- Press button #2 to play file #002 once.
- When playing, pressing any button has no effect.

Required Text in MODE.TXT

None, this is the default mode (DN).

Notes
- The button can be released or held down when playing.
- If the button is held down when the playback finishes, the sound will be played again.
- If both buttons are pressed or held down at the same time, button #1 prevails.

2. Allow interruption during playback.

Intended Operation
- Press button #1 to play file #001 once.
- Press button #2 to play file #002 once.
- Playing can be interrupted by pressing any other button.

Required Text in MODE.TXT

DI

Notes
- When playing, pressing the same button again has no effect.

3. Play the sound only once even if the button is held down.

Intended Operation
- Press button #1 to play file #001.
- Press button #2 to play file #002.
- Don't repeat the sound even if the button is held down.

Required Text in MODE.TXT

DS
N001=F001,J888
N002=F002,J888
H888=J888
END

Notes
- The system will stay in the "H888=J888" endless loop for as long as the button is held down.

4. Play only when the button is held down.

Intended Operation
- Press & hold button #1 to play file #001.
- Press & hold button #2 to play file #002.

Required Text in MODE.TXT

DH

Notes
- Playing stops as soon as the button is released. If the same button is pressed again later, playing re-starts from the beginning of the file instead of where it left off.
- If both buttons are held down, button #1 prevails. To play both sounds alternately when both buttons are held down, use the RH mode.

5. Play different sounds each time when the same button is pressed.

Intended Operation
- Press button #1 to play file #001 the first time, file #002 the second time, and etc.
- Button #2 sequences through file #201, #202...
- When playing, pressing any button has no effect.

Required Text in MODE.TXT

SN

Notes
- Up to 99 files can be assigned to each button and file numbers but be consecutive.
- To allow playback interruption, use the SI mode.
- To play only when the button is held down, use the SH mode. In the SH mode, the same sound will repeat for as long as the button is held down. To advance to the next sound, the button must be released first.

6. EM38A-R: Turn on a single, different relay for each sound.

Intended Operation
- Turn on relay #1 when playing file #001.
- Turn on relay #2 when playing file #002.

Required Text in MODE.TXT

Only the mode letters are required, for example: DH.

7. EM38A-R: Turn on multiple relays for each sound.

Intended Operation
- Press button #1 to play file #007 and turn on relays #1 & 2.
- Press button #2 to play file #008 and turn on relays #2 & 3.

Required Text in MODE.TXT

DS
N001=XN1,XN2,F007,XFF
N002=XN2,XN3,F008,XFF
END

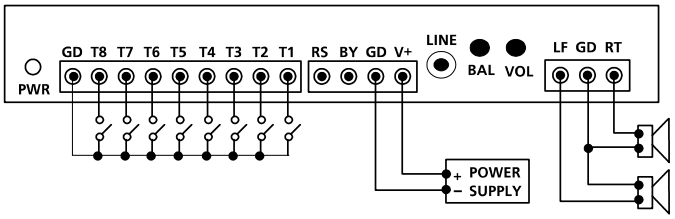
Notes
- XFF is used to turn off all relays.

8. EM38A-X: Pre-program relays for synchronized activations during audio playback.

The relays will be activated automatically when playing QSA files. There is no required text in MODE.TXT.

Application Example

Automatic Playback on Power-up



1. Repeat continuously on power-up.

Intended Operation

- Play file #001 on power-up if switch #1 is closed.
- Play file #002 on power-up if switch #2 is closed.

.....

- Play file #008 on power-up if switch #8 is closed.
- If more than one switch is closed, play all corresponding files sequentially
- Repeat the file(s) until power is turned off.

Required Text in MODE.TXT

RN

Notes

- If priority is required, use DN instead of RN. In this case switch #1 has the highest priority, switch #2 has the second highest priority, and switch #8 has the lowest priority. If more than one switch is turned on, only the file for the switch of the highest priority will be played.

2. Repeat at intervals on power-up.

Intended Operation

- Same as the example above but play files at fixed intervals.
- Only files with corresponding switches closed will play.

Required Text in MODE.TXT

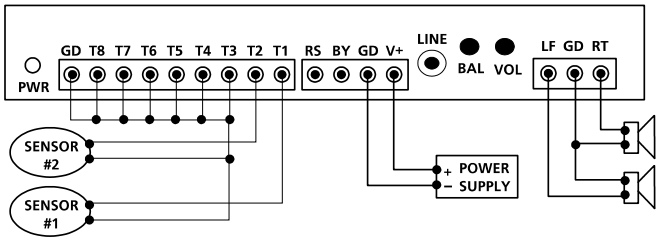
RS
N001=F001,W06000
N002=F002,W06000
.....
N008=F008,W06000
END

Notes

- W06000 is a delay loop of 6000 x 0.1 second = 10 minutes, so the file(s) will be played at an interval of 10 minutes, one file at a time.

Application Example

Using Normally Closed Switches & Sensors



Normally closed switches and sensors are often used in security systems.

1. Play when the switch/sensor opens, with no priorities.

Intended Operation

- Play file #001 when sensor #1 opens.
- Play file #002 when sensor #2 opens.
- Repeat the sound for as long as the sensor is open.
- If multiple sensors are open at the same time, play all corresponding files sequentially.

Required Text in MODE.TXT

RNO

Notes

- Unused inputs must be connected to the ground, as shown in the wiring diagram.

2. Play when the switch/sensor opens, with priorities.

Intended Operation

- Same as the example above, but if multiple sensors are open at the same time, the sensor of the highest priority prevails.

Required Text in MODE.TXT

DNO

Notes

- The sensor connected to T1 has the highest priority, and the sensor connected to T8 has the lowest priority.

3. Play when the switch/sensor opens, but don't repeat.

Intended Operation

- Play file #001 when sensor #1 opens.
- Play file #002 when sensor #2 opens.
- Do not repeat the sound.

Required Text in MODE.TXT

DSB
N001=F001,J999
N002=F002,J999
H999=J999
END

Notes

- The system will stay in the "H999=J999" endless loop for as long as the sensor is open. The system will not respond to other sensors until this one is closed.